

**Evolution of a Distributed Live, Virtual, Constructive Environment for Human in the Loop
Unmanned Aircraft Testing**

James R. Murphy^{*}

NASA Ames Research Center, Moffett Field, CA

Neil D. Otto[†]

Flight Research Associates, Moffett Field, CA

Abstract

NASA's Unmanned Aircraft Systems Integration in the National Airspace System Project has developed a distributed test environment that enables evaluation of the alerting and guidance provided to an unmanned aircraft pilot. The underlying requirement for the test environment was to support human in the loop simulations as well as live aircraft flight testing. To satisfy both, the project leveraged live, virtual, constructive infrastructure concepts to provide a common system architecture. As with any development effort, compromises in the underlying system architecture and design were made to allow for the rapid prototyping and open-ended nature of the research. However, through an incremental build-up approach, the core test infrastructure was implemented to migrate unmanned aircraft detect and avoid algorithm and display concepts developed and tested under simulation into flight test operations with minimal modification. The distributed nature of the test environment enabled efficient testing by leveraging simulation and flight assets from across multiple NASA Centers and other project partner facilities. In addition, using standard live, virtual, constructive capabilities support integration with future research platforms.

Introduction

The National Aeronautics and Space Administration (NASA) is conducting research under the Unmanned Aircraft Systems (UAS) Integration in the National Airspace System (NAS) Project (hereby known as UAS-NAS) to investigate technologies and collect evidence supporting the definition of standards that will enable routine UAS access to the NAS. The UAS-NAS project has two primary technical challenges addressing this access.¹ The first is Command and Control (C2) communications, which includes data required to pilot an aircraft and communicate with air traffic control (ATC) via voice, either within or beyond line of sight with a ground control station (GCS). This capability can be achieved either through a satellite or terrestrial based system. The second technical challenge is Detect and Avoid (DAA); defined as the ability for an UAS to maintain "well clear" of other aircraft, replacing the inherent capability of a manned aircraft pilot to see-and-avoid.²

With guidance from the Federal Aviation Administration (FAA) and industry, the project developed a series of simulations and flight tests designed to collect data to define the C2 and DAA standards.³ In support of these data-gathering activities, the UAS-NAS project created the integrated test and evaluation team to develop the test infrastructure and execute the large-scale integrated events.⁴ Through the project planning effort, it was determined that the C2 simulation and flight test requirements were primarily stand-alone, needing little system integration for human in the loop (HITL) testing. However, the DAA test planning included pilot and air traffic control participants and interaction between the aircraft and the UAS GCS.

Based on the known DAA simulation and flight test requirements, the IT&E team designed a distributed live, virtual, and constructive (LVC) test environment for the underlying system infrastructure. LVC

^{*} Technical Lead, Aerospace Simulation Research and Development Branch

[†] Integration Manager, Aerospace Simulation Research and Development Branch

environments are widely used by the Department of Defense and throughout the aerospace industry.^{5,6,7} The UAS-NAS LVC distributed environment (known as LVC-DE) is comprised of ATC workstations, aircraft simulators, live aircraft, and unmanned aircraft GCSs that, operating together, provide researchers with a relevant NAS environment to test unmanned systems. By modeling the UAS-NAS test system on an LVC paradigm, the project was able to leverage lessons learned from the DoD and industry concepts as well as utilize NASA's existing LVC technologies.⁸ In order to maximize the use of available resources, the LVC test environment was designed to: 1) enable technologies developed by both in-house researchers and external partners, 2) integrate those technologies into the test environment, and 3) distribute the data to local and remote sites. This underscores the two driving requirements for the system, that 1) the LVC must be flexible enough to support the integration of technologies as needed for data collection and 2) that the LVC must support data distribution across NASA facilities to allow for integration of test assets (e.g. ATC facilities, aircraft test ranges) where they were located.

This paper documents the development of the LVC test environment used by the UAS-NAS project for its DAA related simulations and flight tests. It provides a description of the evolution of the underlying LVC infrastructure as it matured from the initial concept into the system to be used for the second phase of the project. Lastly, it documents the plans to migrate the software to enable its integration with future LVC research platforms.

Background

Detect and Avoid

United States Federal Aviation Regulations state that under visual meteorological conditions, whether an aircraft is flying under visual flight rules (VFR) or instrument flight rules (IFR), the pilot has the ultimate responsibility to avoid other aircraft:

"When weather conditions permit, regardless of whether an operation is conducted under instrument flight rules or visual flight rules, vigilance shall be maintained by each person operating an aircraft so as to see and avoid other aircraft. When a rule of this section gives another aircraft the right-of-way, the pilot shall give way to that aircraft and may not pass over, under, or ahead of it unless well clear."⁹

For unmanned aircraft, this rule imposes two problems: 1) "See and avoid" beyond visual line of site (BVLOS) is a challenge since the pilot is not collocated with the aircraft, and 2) What is considered to be "well clear" is left to pilot judgment. The DAA research is attempting to define alternatives to "see and avoid" through sensor technologies and define "well clear" in discrete terms that can be used by algorithms to provide meaningful guidance.¹⁰

Similar concepts for aviation have been defined in the past for collision avoidance (CA). The Traffic Alert and Collision Avoidance System (TCAS) for manned aircraft uses defined separation parameters and negotiation between transponders on equipped aircraft to provide a pilot with commands to climb or descend to avoid an imminent loss of separation. The DAA and CA concepts are intertwined, with DAA enabling pilots to prevent instances of loss of well clear and CA providing pilots with advisories to limit the severity of the loss of well clear and avoid a near mid-air collision. Figure 1 provides a simplified graphic of the DAA and CA relationship with respect to temporal and spatial thresholds. Please refer to Santiago et. al. for a more detailed DAA discussion.¹¹

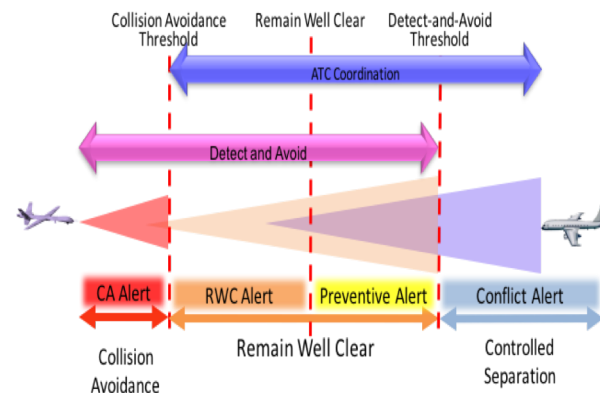


Figure 1. DAA/CA Relationship. *Detect and Avoid helps keep aircraft from losing separation, while Collision Avoidance provides advisories after aircraft lose separation.*

Stakeholders

RTCA (formally known as the Radio Technical Commission for Aeronautics) was chartered by the FAA to operate advisory committees that develop solutions to real-world air transportation problems.¹² In order to safely integrate UAS into non-segregated airspace, the FAA and UAS stakeholders have determined that both a robust DAA concept and a robust and secure C2 data link capability need to be established. In response, the FAA established the Unmanned Aircraft Systems Integration Office to oversee integration of UAS safely and efficiently into the NAS. RTCA formed Special Committee 228 (SC-228) to develop the Minimum Operational Performance Standards (MOPS) for DAA and C2, with emphasis in an initial phase of standards development for civil UAS equipped to operate in Class A airspace flying under IFR.

In parallel, under a separate sub-committee (SC-147), RTCA is developing collision avoidance MOPS based on the Airborne Collision Avoidance System for NextGen (ACAS X) software.^{13,14} The ACAS X_A algorithm (the “A” denotes “active” surveillance) is a proposed replacement for TCAS for manned aircraft and ACAS X_U is the unmanned aircraft variant.

As stated previously, the LVC development was linked closely with the DAA technical research. The LVC-DE test environment described in this paper was developed to facilitate data collection for RTCA SC-228 DAA and SC-147 ACAS X_U MOPS development. The Phase 1 DAA MOPS were released in May 2017 and covered unmanned aircraft transitioning to and from Class A or special use airspace, traversing Class D, E, and G airspace. The MOPS include requirements for air-to-air radar characteristics, requirements for the DAA algorithm, pilot display guidance, and the definition of “Well Clear” as it pertains to its application within the DAA algorithm. The Phase 2 DAA MOPS will build on the standards developed during Phase 1, including extended flight operations in Class D, E, and G airspace, as well as UAS operations in Terminal airspace. In addition, the Phase 2 MOPS will address characteristics of lower SWaP (size, weight, and power) air-to-air sensors and ground based detect and avoid (GBDAA) systems, including whether low SWaP or GBDAA systems will have an impact on the Phase 1 MOPS definition of “Well Clear”. For aircraft that fail to remain well clear and move into the collision avoidance domain, the ACAS X_U MOPS build upon the vertical guidance development of ACAS X_A and incorporate new horizontal collision avoidance logic.

NASA’s UAS-NAS project supports these stakeholders by conducting a series of HITL simulations and unmanned aircraft flight-tests. The purpose of these activities is to collect data and formulate the MOPS requirements to reduce the barriers associated with routine access for unmanned aircraft into national airspace.

LVC Concept

LVC refers to the “live”, “virtual”, and “constructive” systems or components of a test environment. These constructs were first used by the Department of Defense (DoD) to describe the mixing of real and simulated assets. A “live” test component involves human participants operating real systems (e.g. a pilot flying an aircraft). “Virtual” components involve human participants operating simulated systems (e.g. a pilot flying a flight simulator). A “constructive” component is similar to a virtual component, but generally has no interactive human involvement. Instead, the component actions unfold using rule-based decisions (e.g. simulated traffic on a scripted flight path).⁵

A key feature of an LVC environment, is the abstraction of the source of a data feed (e.g. the position of an aircraft) from the client that uses the data. In this way, the user of the data cannot immediately determine whether an aircraft shown on a display is real or simulated. This allows researchers to utilize the same basic system design for testing subjects and collecting data whether the inputs are from live or virtual aircraft.

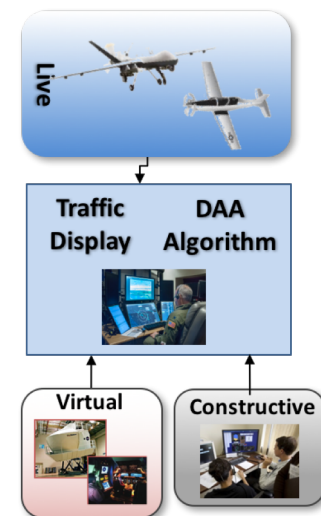


Figure 2. LVC-DE Environment Concept of Operations. An LVC environment promotes the integration of multiple live and virtual data sources.

Using an LVC also supports mixing of live, virtual, and constructive traffic to immerse a test subject in a realistic, but safe test environment.

Figure 2 provides a high-level concept of operations for the LVC-DE test environment developed for project simulations and flight tests under NASA's UAS-NAS project. The LVC-DE enabled the test engineers to integrate existing ATC workstations and simulation infrastructure resident at NASA Ames Research Center with the test aircraft flying in the restricted airspace surrounding NASA Armstrong Flight Research Center (formerly known as NASA Dryden). The underlying LVC infrastructure connects the facilities at the two Centers. The use of abstracted integration through a well-defined interface obscures the source of the data (whether from a live aircraft or virtual flight simulator) used by the ATC and pilot displays as well as DAA algorithms.

While the live, virtual, and constructive components of a test environment only comprise a portion of what is required to run a simulation or flight test, the test environment is widely known as an LVC.[‡] Typical LVC core functionalities are described in the next section.

LVC Core Components

Figure 3 provides a high-level depiction of the core LVC components (shown in blue) developed or used by the LVC-DE. Notice that the LVC core components manage the messaging among the client software systems running in the test environment. These components include the LVC Middleware, Middleware Toolboxes, and the LVC Gateway. Each is described briefly in the following sections. Additional details are provided by Murphy, et al.¹⁵

1. LVC Middleware

LVC middleware can be considered the backbone of an LVC system. It supports a “publish/subscribe” capability, allowing for targeted routing of data through the system among the various client software systems. For instance, published aircraft position data received from a live aircraft or flight simulator would be available for subscription by an appropriate ATC workstation or pilot display via the LVC middleware message handling. The LVC Labs at NASA Ames use the High Level Architecture (HLA) middleware to provide the message routing among the distributed facilities.¹⁶ Other LVC middleware solutions include DoD's Distributed Interactive Simulation (DIS)¹⁷, AviationSimNet¹⁸, Test and Training Enabling Architecture (TENA)¹⁹, and Data Distribution Service (DDS)²⁰. Migration of NASA's LVC-DE core components from HLA to utilize DDS technologies is discussed later in this paper.

2. Middleware Toolboxes

Middleware solutions have a well-defined message interface that must be met by client software packages attempting to connect to the test infrastructure. Middleware toolboxes provide a mechanism to translate the format and content of the client software into the required middleware format. Using a toolbox

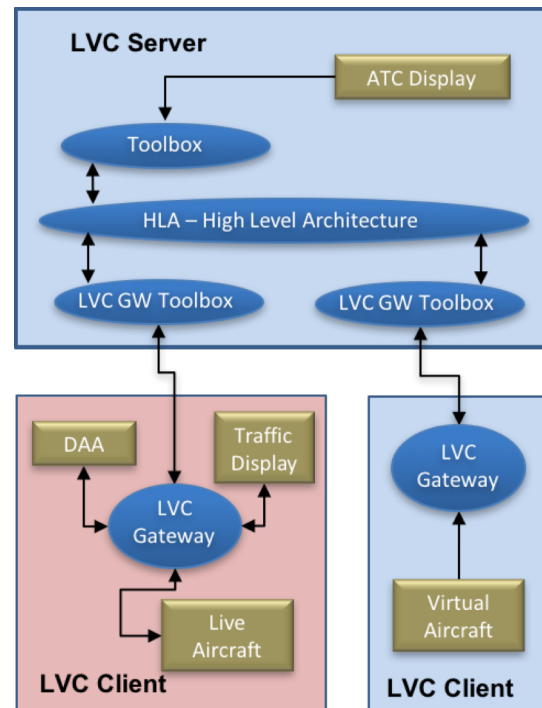


Figure 3. High-level LVC Middleware Connectivity. The usage of LVC middleware to route data through the LVC environment.

[‡] It should be noted that categorizing components of a simulation as live, virtual, or constructive can be problematic. Since the degree of human participation in a simulation is widely variable, as is the degree of equipment realism, there is no clear division between these categories.

instead of implementing the translation directly into the client software has several advantages:

- The development team may not need to control the software in order to implement the interface (i.e. commercial or government off-the-shelf software).
- The message translation may require data transformation or translation calculations that would complicate the client software baseline.
- The software component may connect to multiple different versions of middleware.

However, toolboxes also have a disadvantage for the client software in that a middleware license is always required in order to exchange messages between components. For remote users operating on a limited budget, this can be problematic if the license has significant cost.

3. LVC Gateway

The LVC Gateway was developed to enable connection of client software running at remote facilities. An LVC Gateway process routes local message traffic and provides a single connection from the remote facility to the middleware server (running at NASA Ames in the case of the LVC-DE). Implementing the LVC Gateway to connect remote participating sites addressed two important issues. The first was licensing cost, since no middleware licenses are required to be purchased in order to connect to the LVC. Secondly, the LVC Gateway enabled sites to connect components and test locally without relying on a middleware service to communicate. This feature can be seen in the red shaded box in the lower left corner of Figure 3. The LVC Gateway acts as a local router for messages among the local clients as well as sending data up to the HLA middleware. In addition, multiple LVC Gateway instances can be used to connect separate facilities required for a complete test environment.²¹

LVC Development

The purpose of developing the LVC-DE test infrastructure is to support the data collection needs of the researchers. As such, the development schedule and requirements can be traced to the planned UAS-NAS project activities fairly closely. For convenience, the significant LVC support efforts are divided into four areas detailed in the next sections, namely Initial Capability, Phase 1 DAA MOPS Support, ACAS X_U MOPS Support, Phase 2 DAA MOPS Support.

Initial Capability

The initial effort for the LVC-DE was to develop the LVC core infrastructure and prepare for the early simulations and flight-testing. Prior to receiving specific researcher requirements, high-level architectural system requirements were known. These included the need to ingest both live and virtual data, emulate an en route ATC environment, support a UAS pilot operating from a ground control station, and possibly connect to distributed facilities. While the LVC was envisioned to support the large-scale, distributed simulations and flight test activities, it was quickly realized that integrating the underlying LVC interface into the research test environment would reduce risk with respect to the integration of the DAA and display technologies for those events.

1. LVC Prototype

In order to facilitate integration of the DAA and pilot display via the LVC architecture as early as possible, an LVC prototype system was developed. The LVC developers leveraged the IEEE 1516 standard *Pitch portable Run Time Infrastructure HLA* middleware from NASA's Virtual Airspace Simulation Technology RealTime simulation infrastructure.²² NASA has used the HLA middleware extensively for many years to integrate flight simulators and air traffic control displays into a human in the loop simulation environment. In addition, lessons learned from the development of the Unmanned Aircraft Systems Research Platform—used for unmanned simulations prior to the UAS-NAS project—helped form the basis for the development of the LVC Gateway.⁸ The LVC Gateway was integrated into the pilot display HITL simulations beginning with the first Full Mission simulation, where pilots evaluated early traffic display

features and advisory timing. This was a year prior to its original expected initial use in the first integrated HITL.

2. LVC Characterization

Two of the primary goals of the LVC-DE were to provide a simulation infrastructure that emulates an operational air traffic control environment, as well as a platform to evaluate pilot interaction with DAA advisories and traffic display. As such, the observed latencies in our distributed architecture were characterized to ensure the planned HITL simulation and flight test system architectures fell within known operational ATC latency requirements.²³ Initial LVC environment characterization tests were conducted with two primary objectives:

1. Measure the latency of sending aircraft position updates from the source to the LVC Gateway.
2. Measure the latency of sending aircraft position updates between LVC networked facilities.

A simplified version of the LVC-DE was used to connect the LVC lab at NASA Ames to the ATC Lab at NASA Ames, the LVC lab at NASA Armstrong, and the UAS-NAS Communication Ground Station at NASA Glenn. These data provided a general understanding of the system in terms of its ability to transmit the appropriate data in a timely manner. Figure 4 shows the LVC architecture for the data collection with NASA Glenn during a live flight to understand the latencies between the aircraft and the ground as well as between NASA Ames and NASA Glenn facilities. For a detailed description of the characterization testing and results, please see the LVC-DE Characterization Report.²⁴

Phase 1 DAA MOPS Support

The UAS-NAS project planned an integrated human in the loop simulation and two flight test series to inform the development of the Phase 1 DAA MOPS requirements. The LVC-DE infrastructure was developed to support these activities. This section documents how the LVC-DE evolved to support both HITL simulation and flight test activities.

1. Integrated Human in the Loop Simulation

The integrated human in the loop simulation provided the first opportunity for running a simulation distributed across multiple NASA Centers. The technical goals for the integrated HITL were to: 1) evaluate and measure the effectiveness and acceptability of DAA systems (algorithms and displays) to inform and advise UAS pilots; and 2) evaluate and measure the interoperability and operational acceptability of UAS integration concepts for operating in the NAS. A third Project goal was to characterize the simulation and test environment in order to evaluate the state of the simulation architecture with respect to future UAS research activities.

Figure 5 shows the expanded LVC-DE architecture used to facilitate the integrated HITL testing. Of interest for this discussion is the integration of the DAA algorithm and the Pilot Display in the GCS using the LVC environment. This “loose coupling” between components provided a simple mechanism to enable DAA algorithms or displays to be swapped out as research evolved over the years of the UAS-NAS project. A “tight coupling” between a specific DAA

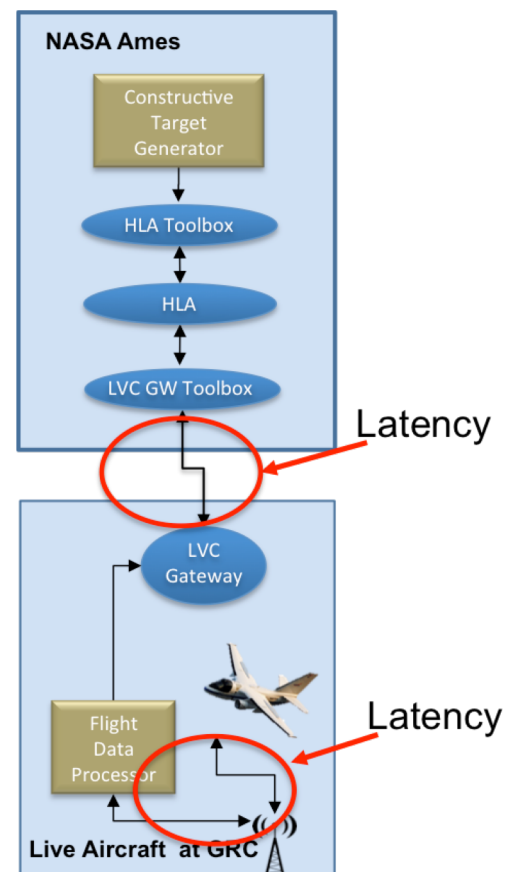


Figure 4. LVC Latency Analysis. An example of the system architecture used to characterize the LVC latencies during live flight.

algorithm and a specific display would have limited flexibility for the researchers. The distributed nature of this architecture enabled the Virtual GCS software (running at NASA Armstrong) to generate and publish the “Ownship”[§] position reports, while the “intruder”^{**} position reports were provided by the Constructive Aircraft Generator running at NASA Ames.

2. DAA Flight Testing

The integrated flight test efforts supported the writing and validation of the draft and final versions of the RTCA SC-228 DAA MOPS. Flight test events were designed to enable collection of data in a realistic operating environment, including the inherent uncertainties of real winds and on-board sensors. However, considering the testing includes the flight of unmanned aircraft, which cannot presently fly in the NAS without restrictions and waivers from the FAA, the integrated test team developed a distributed environment that combines live, virtual, and constructive (or background) traffic in intruder intercept scenarios to promote the safe testing of DAA concepts and technologies.

This combined live/virtual flight test architecture was used during the first integrated flight test (known as Flight Test 3 within the UAS-NAS project). Figure 5 also shows the further expansion of the LVC-DE architecture to incorporate live aircraft. The live UAS sent ownship telemetry and on-board sensor data to the ground to be distributed to client processes via the LVC infrastructure. From the perspective of the ATC participants and the UAS pilot, the “real” live intruder aircraft were indistinguishable from the constructive/virtual intruder aircraft produced by the simulation systems on the ground.

Due to the inherent complexity of conducting live/virtual flight-testing, the flight test setup was ultimately simplified to running scripted encounters solely at NASA Armstrong. To ensure safety, several risk mitigation measures we enacted. These include visual acquisition of the unmanned ownship by the pilot of the manned intruder aircraft prior to 2 nautical miles of separation, an independent source of surveillance provided to the mission test conductor, review of each scripted encounter and expected maneuver with the pilots, and planned abort procedures in case of deviation from of the scripted plan. Researchers were then able to collect targeted encounter based data to test algorithm performance, which was used to validate fast-time and HITL simulations results. The scripted encounter flight-testing setup can be seen the bottom box of Figure 5. In this case no connection to NASA Ames is required (though this link is maintained for system monitoring).

ACAS X_U MOPS Support

As mentioned previously, ACAS X is a NextGen program that is intended to replace and augment the current collision avoidance functionality of TCAS. The unmanned variant, ACAS X_U, adds a horizontal

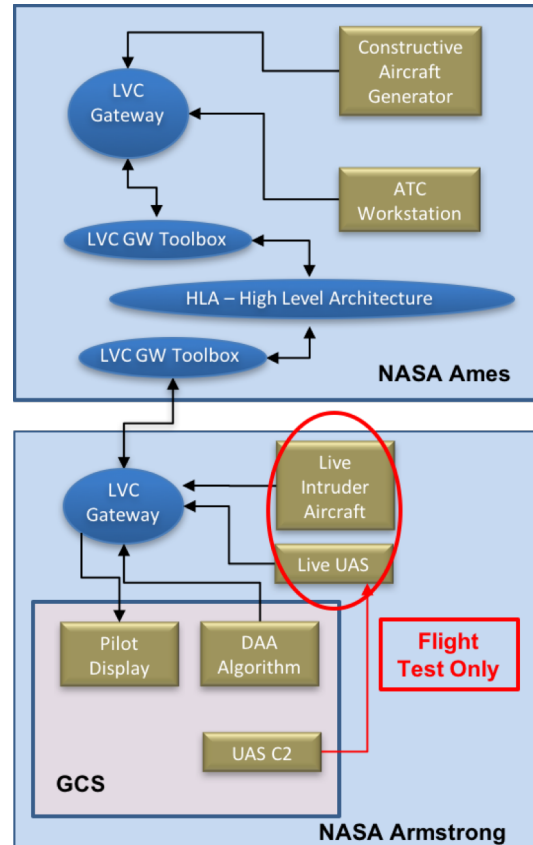


Figure 5. LVC Human in the Loop Simulation and Flight Test Architecture. The integration of client algorithm and display software into the core LVC environment.

[§] “Ownship” is a common reference for the aircraft that is the primary focus of a flight or simulation.

^{**} “Intruder” is a reference to other aircraft that are providing the encounters to the ownship

maneuver capability to the vertical maneuver advisories already built into the “active” ACAS X algorithm under development for manned aircraft. ACAS X_U is also integrating the standards developed under the Phase 1 DAA MOPS to provide a single DAA/CA solution. It is this embedded DAA/CA version of ACAS X_U that is being used for testing using the LVC-DE infrastructure.

1. ACAS X_U Flight Testing

Working closely with the FAA and industry partners from RTCA SC-147, the UAS-NAS project conducted two flight tests of the ACAS X_U algorithm. The first was the initial version of ACAS X_U that had only the vertical CA advisories tailored for the performance characteristics of an unmanned aircraft (ACAS X_U Run 1).²⁵ The second added horizontal CA maneuvers and the a draft of the Phase 1 DAA MOPS algorithm (though only with vertical DAA maneuver guidance, ACAS X_U Run 3).²⁶ Because the ACAS X_U software is run on-board the UAS aircraft in operational tests, the only additional integration into the LVC was to support the ACAS X_U specific advisory messaging. The rest of the LVC system was used without modification.

2. ACAS X_U Software Integration

The ACAS X_U software is being integrated into the LVC environment in order to provide ACAS X_U advisories to pilots during HITL simulations. Because the ACAS X_U software is planned to be incrementally developed with targeted major functional releases, a software container (or wrapper) will abstract the core ACAS X_U algorithm from the interface to the LVC-DE.

Operationally, the ACAS X_U software is run on-board the UAS aircraft, where it inherently receives live sensor data (with real uncertainty). To emulate this, the sensor uncertainty models and ACAS X_U software will be integrated into a single module for both fast-time and human in the loop simulations. Figure 6 shows the LVC architecture encompassing this design, which is intended to more closely represent the operational system and prevent the need for multiple LVC messaging configurations.

Phase 2 DAA MOPS Support

While the UAS-NAS project continues to plan large-scale simulations and flight tests in order to inform the development of the Phase 2 DAA MOPS, the basic LVC system design remains largely constant. LVC development is shifting from the high level architecture changes to integration modifications that ensure long-term scalability and flexibility. This section highlights current efforts aimed at making the LVC design more robust.

1. DAA MOPS messaging

Once the Phase 1 DAA MOPS was finalized, the new DAA messaging requirements were implemented in the LVC infrastructure to support Phase 2 MOPS testing with the goal of ensuring that the LVC supported the standardized DAA alerting and guidance messaging scheme. A secondary goal was to limit the content of the UAS “ownship” and “intruder” aircraft position data messages to only data that would be required in an operational DAA system. This restriction enforced the strict data limitations dictated by the Phase 1 DAA MOPS and ensured that data not available under live flight conditions were not used during simulation. This goal, while well intentioned, posed compatibility problems with the legacy simulation client software.

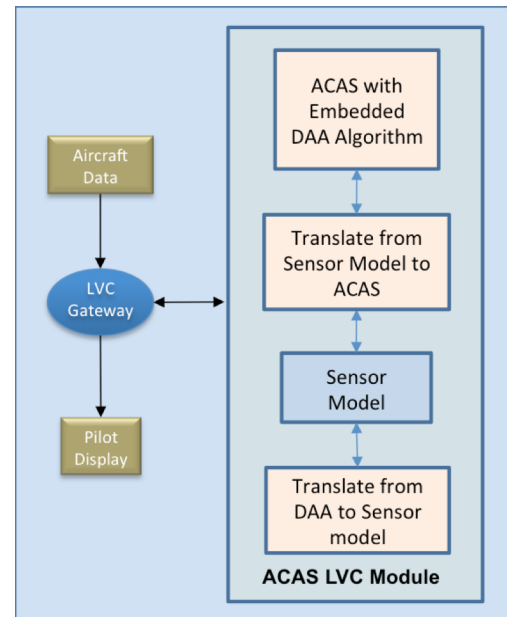


Figure 6. LVC ACAS X_U Simulation Architecture. The integration of the ACAS software and sensor uncertainty model.

Prior to the migration to the DAA messaging for aircraft position reports, the DAA algorithm and UAS pilot traffic display used the legacy aircraft position report message type developed for NASA's ATC simulation displays. This expedited integration of the DAA algorithm and traffic display into the LVC-DE in support of Phase 1 DAA flight-testing and the integrated HITL. However, once the Phase 1 DAA MOPS defined the required data content for the ownship and intruder position messages, the data message requirements for the ground control station and ATC displays diverged. In order to preserve the original LVC design, which utilizes a single data input to provide data for both the GCS and ATC needs, the Surveillance Source Adaptor (SSA) process was developed. SAA translates the DAA position messages into the legacy ATC position messages, or vice versa, depending on the needs and input sources of the LVC environment.

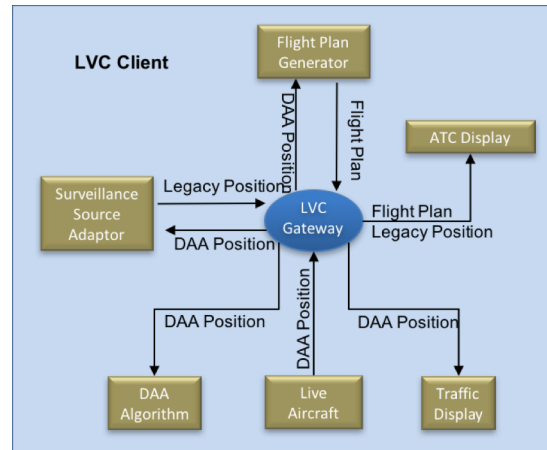


Figure 7. DAA Message Translation. *The LVC connects translation modules to ensure the clients have correctly formatted data.*

In addition, the Phase 1 DAA MOPS does not require flight plan data (e.g. aircraft type, cruise altitude, and beacon code) for the DAA algorithm or pilot display. Flight plans are used by the simulation ATC displays as well as the HLA middleware solution used by NASA. For convenience, software clients providing position reports for a simulation also sent a flight plan message to satisfy this legacy requirement. In order to keep the DAA interface to the LVC as close to the MOPS specification as possible, this additional flight plan message requirement was removed from the client systems connecting to the LVC. To handle aircraft position data with no flight plan information, the Flight Plan Generator (FPG) process was developed. The FPG keeps track of all aircraft in the LVC system and if a position report is received with no corresponding flight plan, a new flight plan message is generated. Figure 7 depicts the translation of the DAA position message into the Legacy position message to support the legacy ATC displays. It also shows the use of the DAA position message by the DAA algorithm and the pilot traffic display.

2. SMART-NAS Integration

NASA is developing a real-time simulation and testing research platform to support investigation of air traffic concepts from regional to national airspace operations.²⁷ This platform, called the Shadow Mode Assessment Using Realistic Technologies for the National Airspace System (SMART-NAS) Test Bed, incorporates state of the art clustering and cloud services capabilities to enable researchers and engineers to test system-level concepts (e.g. traffic flow and terminal scheduling) with live and prerecorded data sets. It also provides a user with a system to manage specific data runs supporting replication of desirable test conditions, or setup of a specific test. The underlying SMART-NAS Test Bed system architecture was developed with inputs from four separate design teams for the purpose of developing a highly scalable and configurable simulation infrastructure. At its core, the Test Bed is a highly configurable LVC infrastructure. This allows it to integrate with the LVC capabilities of the other projects enabling high-fidelity simulation of integrated air traffic management concepts.

The UAS-NAS LVC-DE would benefit from a SMART-NAS Test Bed integration by inheriting the system-wide airspace and decision support tool systems the Test Bed has already subsumed, for example advanced traffic displays, or constructive aircraft generators. Alternatively, the Test Bed would gain access to the UAS aircraft and pilot display interfaces already incorporated into the LVC-DE. Because the SMART-NAS Test Bed was designed using the latest advanced LVC concepts, the UAS-NAS project is investigating the migration of its LVC design to leverage these new concepts. The first step is to migrate the LVC-DE toward using DDS middleware, and away from HLA and the use of LVC Gateways. This would allow the LVC-DE and the Test Bed to immediately exchange messages without the need for toolboxes or message translation processes. In a sense, the LVC-DE would be a new SMART-NAS Test

Bed capability. Figure 8 depicts the SMART-NAS Test Bed system design, incorporating the LVC-DE. A more detailed description of the SMART-NAS Test Bed can be found in Robinson, et. al.²⁸

3. DDS Transition

While the development and use of the LVC Gateway was beneficial during the UAS-NAS Project's maturation period of architecture design and refinement, replacing LVC Gateways with DDS will solve some lingering architecture and design challenges. Much like the LVC Gateway, DDS offers scalable, high performance real-time interoperability, but due to its wide scale industry use it also adds a level of robustness and reliability that the LVC Gateway would never achieve. The built-in networking and data delivery capabilities of DDS will enhance the reliability and adaptability of the architecture. This also allows LVC system designers to focus on the development and management of data topics (the DDS version of messages), which would ultimately replace the LVC Gateway defined interface.

However, the LVC Gateway design considerations applied during its initial development influences how DDS will ultimately be implemented for use by the UAS-NAS Project. One factor is to limit the licensing cost to end users while also enabling them to run, connect, and test system components locally. It's also important to continue to enforce configuration control of data topics just as it was to centrally manage the LVC Gateway interface. The licensing structure for DDS enables this centralized control and development, allowing end users to incorporate topics and connect at no cost. Centralized design of DDS data topics will ease the burden on software component developers by providing them with completed topics that can be compiled into their code, quickly enabling them to connect to the LVC. However, there are still open questions regarding whether the migration from DDS and away from the NASA developed LVC Gateway will limit some long term system flexibility.

Conclusions and Next Steps

The development of the LVC-DE system was based on the UAS-NAS project's research and test system requirements. The LVC-DE has demonstrated its scalable and extensible capabilities throughout Phase 1 DAA MOPS development from the initial project simulations to the flight test encounters. As the project transitions to Phase 2 DAA MOPS data collection, significant message changes are not anticipated, but future LVC implementations will incorporate Phase 1 DAA MOPS guidance as well as focus on eliminating architectural differences between simulation and live flight. Doing so will improve the ability of the researchers to more easily run their applications in both simulation and live environments to take full advantage of the benefits each can provide.

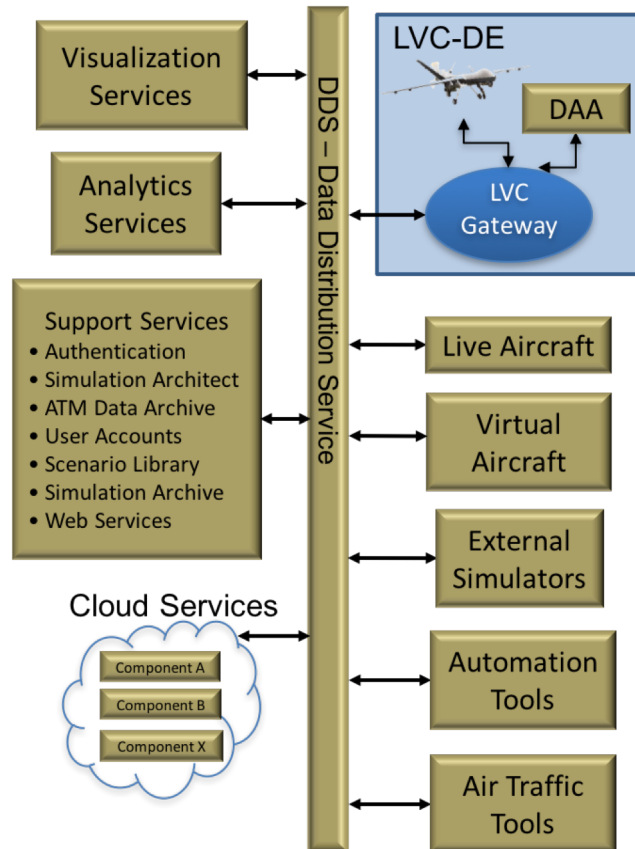


Figure 8. SMART-NAS Test Bed System Design. *The SMART-NAS Test Bed connects clients through a DDS middleware solution, which the LVC-DE is migrating towards.*

As the LVC-DE core development continues to wind down, potential integration with the SMART-NAS Test Bed will extend the UAS-NAS LVC capabilities to other domains. The migration of the core LVC-DE infrastructure from HLA and the internally developed LVC Gateway to the DDS middleware should extend the LVC-DE utility beyond the UAS-NAS project. This transition is a significant step that must first be measured before it is taken to ensure it keeps the UAS-NAS and future projects using the LVC-DE technologies on the path to success.

Looking toward future uses for the LVC-DE, its very definition provides direction for use cases that naturally result from its core capabilities, i.e. the mixing of live and virtual system components. A simplified definition of LVC can be thought of by the following rules:

- **Live:** Real People operating real assets
- **Virtual:** Real people operating simulated assets
- **Constructive:** Simulated people operating simulated assets.

Looking at Table 1, this leaves out a logical extension that is taking place across the aviation industry: “simulated people operating real assets”. Although there are recognized problems in using the term “Autonomous” to describe algorithm control of real assets, varying levels of autonomy and algorithm integration can be tested in all aspects of the Live, Virtual, and Constructive elements of an LVC. Given that autonomy is not restricted solely to simulated people and real assets, using an LVC (or now LVCA) environment to baseline automation concepts can benefit from using a systematic, build-up approach.

Table 1. Simplified LVC(A)

	Real Assets	Simulated Assets
Real People	Live	Virtual
Simulated People	Autonomous	Constructive

References

- ¹ National Aeronautics and Space Administration, NASA Technology Development Project Plan, Unmanned Aircraft Systems (UAS) Integration in the National Airspace System (NAS), 31 January 2013.
- ² Federal Aviation Administration, “Sense and Avoid (SAA) for Unmanned Aircraft systems (UAS),” FAA Sponsored SAA Workshop, October 9, 2009.
- ³ RTCA, “Terms of Reference, RTCA Special Committee 228, Minimum Performance Standards for Unmanned Aircraft Systems”, RTCA Paper No. 109-13/PMC-1089, May 20, 2013.
- ⁴ Murphy, J. R., Hayes, P. S., Kim, S. K., Bridges, W., and Marston, M., "Flight Test Overview for the UAS Integration in the NAS Project," AIAA Paper 2016-1756, January 2016.
- ⁵ DoD: “Modeling and Simulation Master Plan”, DoD 5000.59P, Oct 1995.
- ⁶ Henninger, Amy E., Cutts, Dannie, Loper, Margaret, et al, “Live Virtual Constructive Architecture Roadmap (LVCAR) Final Report”, Institute for Defense Analysis, Sept, 2008.
- ⁷ Bezdek, W. J., Maleport, J., Olshon, R., “Live, Virtual & Constructive Simulation for Real Time Rapid Prototyping, Experimentation and Testing using Network Centric Operations,” AIAA 2008-7090, AIAA Modeling and Simulation Technologies Conference and Exhibit, August 2008.
- ⁸ Lutz, R., LeSueur, K., Fast, P., Graeff, R., Simolte, A., Rutledge, J., and Motti, R., “A Persistent LVC Simulation Environment for UAS Airspace Integration,” The Interservice/Industry Training, Simulation & Education Conference (I/ITSEC), December 2010.
- ⁹ United States Code of Federal Regulations, Title 14, §91.113.

- ¹⁰ Consiglio, M., Chamberlain, J., Munoz, C., Hoffler, K., "Concept of Integration for UAS Operations in the NAS", 28th International Congress of the Aeronautical Sciences, September 2012.
- ¹¹ Santiago, C., Mueller, E., R., "Pilot Evaluation of a UAS Detect-and-Avoid System's Effectiveness in Remaining Well Clear", 10th USA/Europe ATM R&D Seminar (ATM2015), Lisbon, Portugal, 23-26 June 2015.
- ¹² RTCA website: <http://www.rtca.org>, June 2017.
- ¹³ Kochenderfer, M. J. and Chryssanthacopoulos, J. P., "Robust Airborne Collision Avoidance through Dynamic Programming", Massachusetts Institute of Technology, Lincoln Laboratory, 2011, Project Report ATC-371.
- ¹⁴ RTCA, "Terms of Reference, RTCA Special Committee (SC) 147, Aircraft Collision Avoidance Systems", RTCA Paper No. 79-16/PMC-1478, March 17, 2016.
- ¹⁵ Murphy, J. Jovic, S., Otto, N., "Message Latency Characterization of a Distributed Live, Virtual, Constructive Simulation Environment," AIAA Infotech@Aerospace Conference, January 2015.
- ¹⁶ Software website: <http://www.pitch.se/prti>, 25 June 2017.
- ¹⁷ "1278.1-2012 IEEE Standard for Distributed Interactive Simulation - Application Protocols", IEEE, Retrieved 24 June 2017.
- ¹⁸ AviationSimNet Standards Working Group, "AviationSimNet Specification", Version 2.2, MITRE Corporation, June 2010.
- ¹⁹ Edward T. Powell, Jason Lucas, Kurt Lessmann, and George J. Rumford, "The Test and Training Enabling Architecture (TENA) 2002 Overview and Meta-Model," published in the Proceedings of the Summer 2003 European Simulation Interoperability Workshop, 03E-SIW-041, found at <http://www.sisostds.org/>.
- ²⁰ "Data Distribution Service (DDS), Version 1.0". *Object Management Group. December 2, 2004.* Retrieved June 25, 2017.
- ²¹ Murphy, J. Jovic, S., Otto, N., "Message Latency Characterization of a Distributed Live, Virtual, Constructive Simulation Environment," AIAA Infotech@Aerospace Conference, January 2015.
- ²² Lehmer, R., and Malsom, S., "Distributed System Architecture in VAST-RT for Real-Time Airspace Simulation," American Institute of Aeronautics and Astronautics (AIAA), 2004.
- ²³ FAA: "NAS System Specification, Functional and Performance Requirement for the National Airspace System, General, Volume 1," FAA, Department of Transportation, NAS-SS-1000, 15 April 1995.
- ²⁴ NASA: "Live, Virtual, Constructive Distributed Test Environment: Characterization Report," NASA UAS-04.05.0001.01, September, 2013.
- ²⁵ FAA: "ACAS XU FTP, Version 2, Revision 1", October 29, 2014.
- ²⁶ Carey, B., "ACSS Among Teams Advancing Collision Avoidance for Unmanned Aircraft", Retrieved from www.aionline.com, 4 August 2017.
- ²⁷ Palopo, K., Chatterji, G., Guminsky, M., Glaab, P., "Shadow Mode Assessment using Realistic Technologies for the National Airspace System (SMART NAS) Test Bed Development," AIAA Aviation, June 2015.
- ²⁸ Robinson, J. E., Lee, A., Chok, F. L., "Development of a High-Fidelity Simulation Environment for Shadow-Mode Assessments of Air Traffic Concepts," Royal Aeronautical Society Modelling & Simulation in Air Traffic Management Conference, November 2017.